AMENDMENTS TO THE CLAIMS

This listing of Claims will replace all prior versions, and listings, of Claims in the Application.

Listing of Claims:

Claim 1 (Currently amended): A method for determining the integrity of an application program running under an operating system on a computer system having a memory, ~~said application program having at least a data portion residing in the memory,~~ the method comprising the steps of:

(a) ~~pre-~~allocating one or more segments in the memory for at least a ~~said~~ data portion of the application program;

(b) inserting tables in said data portion ~~segments~~;

(c) building the ~~executing said~~ application program on said computer system ~~using an operating system, said application program produced~~ by the steps of:

(c1) providing a linker operable to associate addresses across relocatable modules and further operable to output relocation data stored in said relocatable modules;

(c2) ~~(c1)~~ linking via said linker one or more relocatable object modules with one or more libraries and other object modules to form an intermediate executable module, said relocatable object modules being pre-

compiled, and said libraries and said other object modules ~~comprising~~ including said relocation data~~,~~;

(c3) ~~(c2)~~ selecting addresses of portions of said libraries and said other object modules linked to said intermediate executable module by examining during said linking step said relocation data output by said linker ~~to determine selected addresses, said selected addresses corresponding to address locations in said segments,~~;

(c4) ~~(c3)~~ storing said selected addresses in said tables~~,~~;

(c5) ~~(c4)~~ storing a default address of a selected subprogram of said intermediate executable module in said data portion~~,~~; and

(c6) ~~(c5)~~ loading said libraries and said other object modules in said memory to transform said intermediate executable module into ~~said~~ the application program ~~executable by said computer system~~;

(d) executing the application program under the operating system of the computer system, said tables being retained in said at least one data portion during said application program execution;

(e) ~~(d)~~ determining a reference address associated with said selected subprogram at any time after the ~~run-time for said~~ application program begins execution;

(f) ~~(e)~~ comparing said reference address with said default address; and,

(g) ~~(f)~~ executing a security application or module to verify the integrity of the application program at addresses determined from ~~to determine said integrity of said application program based on~~ said reference address and said selected addresses in said tables.

Claim 2 (Currently amended): The method of claim 1, wherein said addresses determined in said security application or module executing step (g) ~~(f) uses~~ is said reference address if said reference address is equal to said default address.

Claim 3 (Currently amended): The method of claim 1, further ~~comprising~~ including the step of computing a substitute address by offsetting memory locations of said selected addresses stored in said tables for every selected address.

Claim 4 (Currently amended): The method of claim 3, wherein said derived address in said security application or module executing step (g) ~~(f) uses~~ is said substitute address if said reference address is unequal to said default address.

Claim 5 (Currently amended): The method of claim 3, ~~wherein said selected addresses are offset~~ further including the step of updating said tables by adding or subtracting an offset value to said selected addresses stored therein.

Claim 6 (Original): The method of claim 1, wherein said selected addresses are selected from a group consisting of memory references and jump target addresses, and said subprograms are selected from a group consisting of functions, subroutines, procedures and libraries.

Claim 7 (Currently amended): The method of claim 1, wherein said security application ~~is~~ computes a checksum ~~application~~.

Claim 8 (Currently amended): The method of claim 1, wherein said security application executes a decryption algorithm on ~~decrypts~~ previously encrypted data.

Claim 9 (Currently amended): The method of claim 8, ~~wherein said data is encrypted while said tables are being inserted~~ further including the step of encrypting at least one code segment or data segment of the application program simultaneously with said table inserting step (b).

Claim 10 (Currently amended): The method of claim ~~1~~ 9, wherein said application program ~~comprises~~ includes encrypted data residing on a ~~DVD~~ digital versatile disk.

Claim 11 (Currently amended): A computer system, comprising:

a central processing unit;

memory accessible by the central processing unit;

a program code translator for building an application program, said program code translator executable on said central processing unit and within said memory and including an application program loader and a program code linker, said linker operable to associate addresses across relocatable software modules and further operable to output relocation data stored in said relocatable modules;

at least one application program built by said program code translator and executable on said central processing unit and within said memory, said application program including at least one data segment having formed therein at least one address table, said address table having stored therein selected addresses of program code from one or more relocatable object modules as linked together by said linker, said selected addresses determined from said relocation data stored in said one or more relocatable object modules as output by said linker; and,

a security application executing a security algorithm on said program code at addresses determined from a reference address of a previously selected subprogram of said program code and said relocation data stored in said table, said security application operable to access said table while said application program is executing ~~means for determining the integrity of said application program~~

~~according to a method as described in claim 1~~.

Claim 12 (Currently amended): The computer system of claim 11, wherein said ~~step of executing a~~ security application ~~uses~~ is operable to access said program code at memory locations based solely on said reference address if said reference address is equal to ~~said~~ a default address of said previously selected subprogram.

Claim 13 (Currently amended): The computer system of claim 11, ~~further comprising the step of computing~~ wherein said central processing unit is operable to compute a substitute address for every said selected address stored in said table by offsetting memory locations of said selected addresses stored therein ~~in said tables for every said selected address~~.

Claim 14 (Currently amended): The computer system of claim 11, further ~~comprising the step of storing~~ including a data compressor/encryptor operable to store said selected addresses in a compressed and encrypted format in said tables.

Claim 15 (Currently amended): The computer system of claim 13, wherein said ~~step of executing a~~ security application is operable to access said program code at memory locations based on ~~using~~ said substitute address if said reference

address is unequal to ~~said~~ a default address of said previously selected subprogram.

Claim 16 (Currently amended): The computer system of claim ~~11~~ 29, wherein said application program ~~comprises~~ includes encrypted data residing on a ~~DVD~~ digital versatile disk.

Claim 17 (Currently amended): A computer readable medium having computer-executable instructions, which when executed on a computer system, causes said computer system to determine the integrity of an application program running under an operating system on said computer system, ~~said application program having at least a data portion residing in memory~~, said computer-executable instructions causing said computer system to perform the steps of:

(a) ~~pre~~-allocating one or more segments in the memory for at least a ~~said~~ data portion of the application program;

(b) inserting tables in said data portion ~~segments~~;

(c) building the ~~executing said~~ application program on said computer system ~~using an operating system, said application program produced~~ by:

(c1) linking via a program code linker one or more relocatable object modules with one or more libraries and other object modules to form an intermediate executable module, said relocatable object modules being

pre-compiled, and said libraries and said other object modules ~~comprising~~ including relocation data, said program code linker operable to output said relocation data;

(c2) selecting addresses of portions of said libraries and said other object modules linked to said intermediate executable module by examining during said linking step said relocation data output by said program code linker ~~to determine selected addresses, said selected addresses corresponding to address locations in said segments,~~;

(c3) storing said selected addresses in said tables~~,~~;

(c4) storing a default address of a selected subprogram of said intermediate executable module in said data portion~~,~~; and

(c5) loading said libraries and said other object modules to transform said intermediate executable module into ~~said~~ the application program ~~executable by said computer system~~;

(d) executing the application program under the operating system, said tables being retained in said data portion during said application program execution;

(e) ~~(d)~~ determining a reference address associated with said selected subprogram at any time after the ~~run-time of said~~ application program begins execution;

(f) ~~(e)~~ comparing said reference address with said default address;

and,

(g) ~~(f)~~ executing a security application or module to verify the integrity of the application program at addresses determined from ~~determine said integrity of said application program based on~~ said reference address and said selected addresses in said tables.

Claim 18 (Currently amended): The computer readable medium of claim 17, wherein the computer-executable instructions stored thereon causes said computer system to further ~~comprising~~ perform the step of computing a substitute address by offsetting memory locations of said selected addresses stored in said tables for every said selected address, said step of executing a security application being based on using said substitute address if said reference address is unequal to said default address.

Claim 19 (Currently amended): The computer readable medium of claim 17, wherein addresses determined in said security application or module executing said step (g) ~~(f)~~ is ~~based on using~~ said reference address if said reference address is equal to said default address.

Claim 20 (Currently amended): A method for determining the integrity of a relocatable application program executable on a computer system, ~~said~~ the application program being generated from one or more pre-compiled object files, ~~said~~ the computer system including memory and ~~said~~ the application program having at least a data space residing in ~~said~~ the memory, ~~said~~ the method comprising the steps of:

(a) inserting tables into pre-allocated memory segments residing in ~~said~~ the data space;

(b) examining relocation data for selected addresses when ~~said~~ the pre-compiled object files are linked by a program code linker ~~and loaded~~ with one or more libraries and other object files and loaded into the memory, said program code linker operable to output relocation data stored in relocatable object modules and libraries, said libraries and said other object files ~~comprising~~ having said relocation data stored therein, said relocation data examination being of said relocation data stored in said pre-compiled object files and said one or more libraries upon output by said program code linker;

(c) storing said selected addresses in said tables;

(d) storing a default address in ~~said~~ the data space, said default address being associated with a point of reference within ~~said~~ the pre-complied object files, said libraries and said other object files, said tables and said default address being retained in the data space until the application program is unloaded

from the memory;

(e) determining a reference address from ~~said~~ the application program at run-time, said reference address corresponding to said point of reference;

(f) comparing said reference address with said default address; and

(g) performing a checksum to determine ~~said~~ the integrity of ~~said~~ the application program at addresses determined from ~~based on~~ said reference address and said selected addresses in said tables.

Claim 21 (Currently amended): The method of claim 20, wherein said step of performing a checksum is based on using said reference address to access the application program if said reference address is equal to said default address.

Claim 22 (Currently amended): The method of claim 20, wherein said step of performing a checksum is based on using a substitute address to access the application program if said reference address is unequal to said default address, said substitute address being computed by offsetting memory locations of selected addresses for every selected address.

Claim 23 (Original): The method of claim 22, wherein said offsetting is done by subtraction or addition.

Claim 24 (Currently amended): The method of claim 20, wherein said application program ~~comprises~~ includes encrypted data residing on a ~~DVD~~ digital versatile disk.

Claim 25 (Currently amended): A computer readable medium having computer-executable instructions, which when executed on a computer system may encrypt and/or decrypt a portion of an application program enabled to run under an operating system on ~~said~~ the computer system, and causes ~~said~~ the computer system to determine the integrity of ~~said~~ the application program ~~having at least a data portion~~ residing in memory, ~~said~~ the computer-executable instructions causing ~~said~~ the computer system to perform the steps of:

(a) ~~pre~~-allocating one or more segments in the memory for at least a ~~in said~~ data portion of the application program;

(b) inserting tables in said data portion ~~segments~~;

(c) building the ~~executing said~~ application program on said computer system ~~using an operating system, said application program produced~~ by:

(c1) linking via a program code linker one or more relocatable object modules with one or more libraries and other object modules to form an intermediate executable module, said relocatable object modules being pre-compiled, and said libraries and said other object modules ~~comprising~~

including relocation data, said program code linker operable to output said relocation data;

(c2) selecting addresses of portions of said libraries and said other object modules linked to said intermediate executable module by examining during said linking step said relocation data output by said program code linker ~~to determine selected addresses, said selected addresses corresponding to address locations in said segments,~~;

(c3) storing said selected addresses in said tables~~,~~;

(c4) storing a default address of a selected subprogram of said intermediate executable module in said data portion~~,~~; and,

(c5) loading said libraries and said other object modules to transform said intermediate executable module into ~~said~~ the application program ~~executable by said computer system~~;

(d) modifying one or more portions of said ~~segments~~ data portion by encryption;

(e) executing the application program under the operating system, said tables being retained in said data portion during said application program execution;

(f) ~~(e)~~ determining a reference address associated with said selected subprogram at run-time of ~~said~~ the application program;

(g) ~~(f)~~ comparing said reference address with said default address;

and

(h) ~~(g)~~ executing a security application or module to verify the integrity of the application program at addresses determined from ~~determine said integrity of said application program based on~~ said reference address and said selected addresses in said tables.

Claim 26 (Currently amended): The computer medium of claim 25, wherein the computer-executable instructions cause the computer system is caused to prevent ~~said method prevents~~ access to encryption keys associated with ~~said~~ the application program.

Claim 27 (Currently amended): The computer medium of claim 25, wherein said step of modifying one or more portions of said segments by encryption ~~comprises the step~~ includes the steps of:

(a) adding or subtracting an offset value to memory locations of said selected addresses in the encrypted object;

(b) performing decryption; and

(c) adding or subtracting said offset value to said memory locations of said selected addresses after decryption.

Claim 28 (New): The method of claim 11, wherein said security algorithm includes a checksum computing algorithm.

Claim 29 (New): The method of claim 11, wherein said security algorithm includes a decryption algorithm.